Forecasting stock return using a recurrent neural network apply to a financial optimization problem

Student: Juliana Ochoa Ramírez
Advisor: Paula Maria Almonacid Hurtado
Maestría en ciencias de los datos y analítica
Universidad EAFIT

May 24, 2021

Abstract

This paper presents a methodological proposal for optimizing financial asset portfolios by incorporating the returns predictions instead of the historical returns to calculate an efficient frontier. We changed the return means methodology to forecast by the return with LSTM neural network. We performed several simulation exercises to evaluate the methodology with real data from the US stock market to examine our portfolio optimization model. To evaluate our results, we compared the mean-variance frontier efficiency with the neural network return model. We selected one optimal portfolio that offered the highest expected return for a defined level of risk and compare both models. We show how the neural network return model has a better performance for different periods of time, outperforming the mean-variance model at the same level.

Keywords: Neural Network, LSTM, forecasting, portfolio optimization

1 Introduction

The portfolio allocation process has different investment horizons, such as short-term, medium-term, and long-term. This work will focus on the short term in a stock market like that of the United States. One of the techniques most used has been the Markowitz model because it is easier to implement in different advisor companies. It selects the most efficient portfolio by analyzing various possible portfolios of the given securities (Markowitz, 1952). We need to find three factors, such as return and risk of each stock and their correlation. Improving stock return forecasting is essential because every investor wants a portfolio that minimizes risk while maximizing returns. It is nearly impossible to build a portfolio entirely devoid of risk due to the stock market's inherent risk. However, potential returns can balance a portfolio or an investment's risk.

We use a recurrent neural network for the development of this project. We develop a neural network for each stock separately, followed by a standard feed-forward output layer and an activation function for each gate. We predict stock returns with the neural network so that it helps in the optimization model. In this case, we find whether this new forecasting effectively overcomes Markowitz's efficient frontier results. Stock returns will be the principal difference between both models.

Therefore, finding an optimal portfolio using an efficient frontier and changing the methodology to stock return is the purpose of this study. This work has several sections. Following this introduction, Section 2 is the literature review of all authors who have written about this topic. Section 3 presents a short review of the data. Section 4 introduces the method and describes it in a general sense. Section 5 evaluates the methodology, and the last section outlines the conclusions and future work.

2 Literature review

Portfolio selection is a resource allocation problem in both economic and finance areas. Allocating a set of resources among a group of assets (Po$_{C}$*hang &Pin*$_{C}$*heng*, 2008) is a problem; portfolio optimization is the process of selecting the best portfolio out of all portfolios considered, according to some objective. For this reason, " *the mean-variance model proposed by Harry Markowitz is a landmark in modern portfolio theory*" (Markowitz, 1952). This model minimizes the risk of investment in a portfolio of stocks by optimizing stocks with low joint risks. The dual approach maximizes the portfolio's return at a given risk, which provides a mechanism for loss compensation known as efficient diversification.

Markowitz's model has the assumption that the time series of returns of each stock follows a normal distribution and uses its mean as a prediction of the stock's future return. Its variance is measured by the stock's risk and each time series pairs' covariance to measure each pair of stocks' mutual risk. In this paper, we use this approach like our Navy model.

Nevertheless, the use of mean returns as a prediction of a stock's future returns imposes a filtering effect on the stock market's dynamic behavior, leading to imprecise estimates of short-term future returns (Freitas, De Souza, & De Almeida, 2009). When we invest in a portfolio, we assume the efficient market hypothesis, which implies a random walk model for stock prices, but pricing irregularities and predictable patterns such as serial correlations and calendar effects cause the stock to present a different behavior (Malkiel, 2003).

However, in recent decades, neural networks and other machine-learning algorithms have been successfully forecasting different non-linear models. The key to their success is that given an extensive representative data set, machine-learning algorithms can learn to identify intricate non-linear patterns and explore unstructured relationships without hypothesizing them a priori (Smyl, 2020). Machine-learning methods for time-series prediction have appeared, ranging from neural network models to support vector machines and fuzzy sets theory (White, 1988),where the non-linear mapping capabilities and the non-assisted estimation of the structural model's neural networks' parameters are advantageous for its application in prediction (Hansen & Nelson, 1997).

In literature, many scholars have explored ways to predict stock prices with neural networks. For example, with prior knowledge and a neural network, Kohara and Fukuhara used a neural network to predict stock prices (Kohara, Ishikawa, Fakuhara, & Nakamura, 1997). Heaton and Polson used as well neural networks (Heaton, Polson, & White, 2016). Additionally, extreme machine learning was applied to predicting stock prices by Li and Xie (Li, Xie, & Wang, 2016), and in 2017, there was an applied econometric approach using machine learning (Mullainathan, 2017)).

In the last two years, machine learning has been used, for instance, in pattern graphs by Jeon and Hong (Jeon, Hong, & Chang, 2018); Lee demonstrated convolutional neural networks (Lee, Ahn, Kwahk, & Ahn, 2018) and recurrent neural networks were used to predict stock prices (Graves, Mohamed, & Hinton, 2013), which could be used to find return prices instead of just find the price. Return is calculate is calculated by subtracting the initial value of the stock price from the final value of the stock price.

A recurrent neural network (RNN) is a class of advanced artificial neural network (ANN) that involves directed cycles in memory. Fisher and Krauss demonstrated that an LSTM network could effectively extract meaningful information from noisy financial time series data (Fisher & Krauss, 2017).

According to the literature review, advances in the forecasting time series through deep learning techniques are significant. We seek to implement an LSTM neural network to predict the US stock returns (Freitas et al., 2009) and calculate frontier efficiency. We aim to find the best portfolio through the efficient frontier for a level of risk (portfolio optimization) and compare these results with the mean-variance model. We will be testing for two months in different time-window sizes, and the portfolio will be rebalanced each week.

3 Data

We used the United States' stocks to build an efficient frontier. The data is the stock market's value for business days; particularly in this work, we took just Wednesday´s data. The stock market data was extracted from Yahoo Finance for four stocks, which is all public information. In this case, we use a small data to understand the better behaviour and what would help us better. But, we search to be in a comprehensive market so we select different sectors, nevertheless we use a few stocks to understand better our work.

We selected four different sectors (financial, energy, healthcare, and technology), aiming to have good diversification in the portfolio. Thus, we selected stocks with an actual weight in each index: JP Morgan (*JPM Data*, n.d.) in the financial sector, Chevron Corporation (*CVX Data*, n.d.) in the energy sector, Merck Co. (*MRK Data*, n.d.-a)) in the healthcare sector, and Microsoft Corporation (*MRK Data*, n.d.-b) in the technology sector. Figure 1 shows the behavior of each stock's historical prices.

The data was extracted between 2014 and 2020 and included all the information available; we did not omit anything. Our goal is to predict the values based on the historic values of the companies. Thus, it is essential to understand each stock's return and volatility in the past and at different time windows. Additionally, we plotted returns (Figure 11 in Appendix I) and tried to have a better understanding of the series. During this period, the United States elected Donald Trump as president in 2016. All stocks showed increases in prices from this year until 2019 when the Covid-19 crisis caused prices to increase their levels of volatility. Thus, we see different behavior with more volatility in 2020.

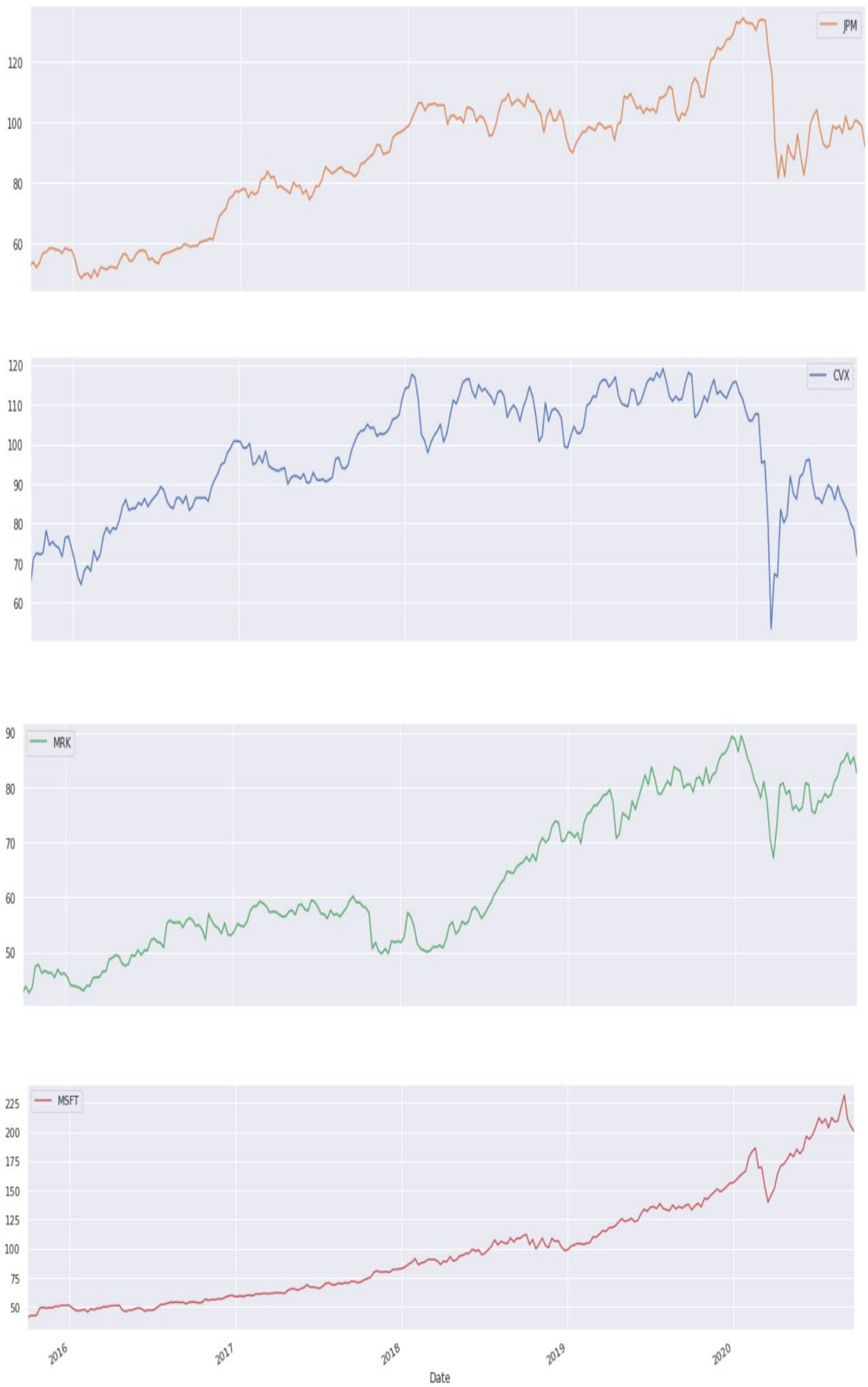


Figure 1: Stock's Prices

4 Methodology

We chose a Long Short-Term Memory (LSTM) (Gers, Schmidhuber, & Cummins, 1999) network to forecast stock prices, a modified version of recurrent neural networks that makes it easier to remember past data in memory. Additionally, it could effectively extract meaningful information from noisy financial time series data (Fisher & Krauss, 2017). Such recurrent neural network characteristics work better for predicting time-series data in the M4 forecasting competition(Smyl, 2020).

To forecast future stock prices of different assets and improve the allocation process, we proposed an architecture whereby we develop a neural network for each stock separately with two LSTM layers, followed by a standard feed-forward output layer and an activation function for each gate. Next, we developed an ensemble neural network from the outputs of the four stocks' neural networks. We concatenated all outputs of the LSTM into dense layers. We used the architecture shown in Figure 2.

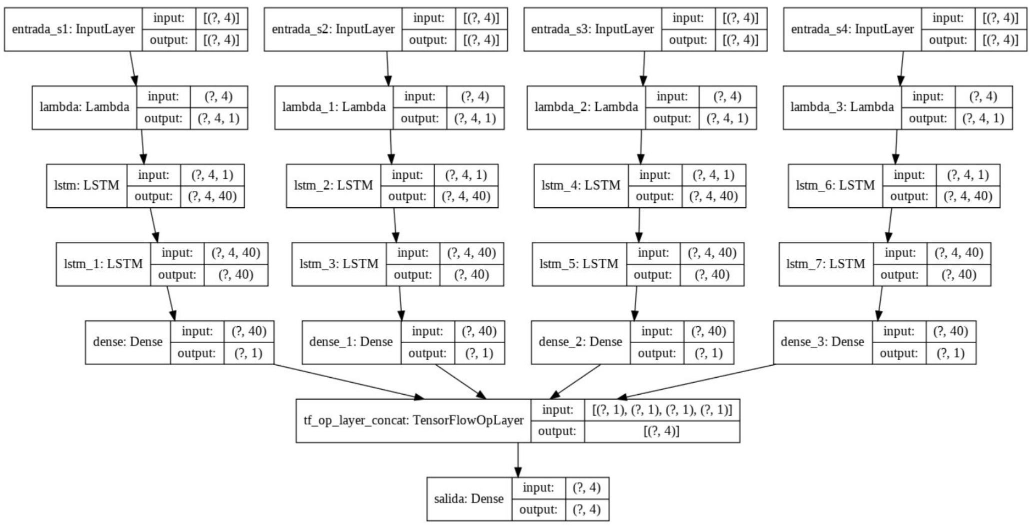


Figure 2: Architecture Neural Network

This architecture gives an output shape of the four stocks with a non-linear correlation and learns different patterns of individual behavior in each stock. The Long Short-Term Memory (LSTM) layers capture the series' temporal and significant history for predictions. The long-term memory usually calls the cell state because this allows the storage of information from previous intervals within the LSTM cell. The cell state is modified by the forget gate placed below the cell state and adjusted by the input modulation gate. The previous cell state forgets by multiplying with the forget gate and adds new information through the input gate's output. The forget gate is the remember vector; the save vector is usually called the input gate. These gates determine which information should enter the cell state for long-term memory. However, the significant parts are the activation functions. We used LSTM cell shown in Figure 3.

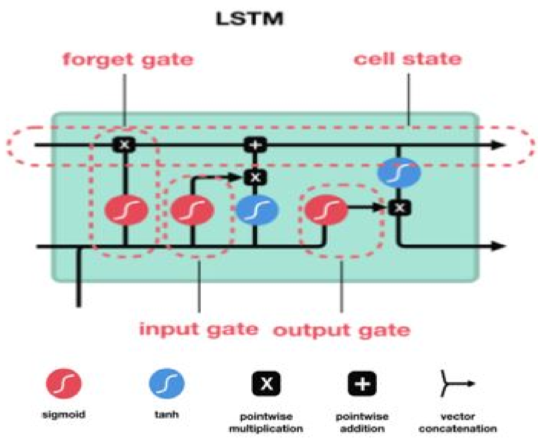


Figure 3: LSTM Cell - Visualization for: (Phi, 2018)

The neural networks need three inputs: first is window size, target size and bath size that is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. Where the inputs used are window size 4, batch size 16, and target size 1. To improve the neural network learning process, we standardized stock price with their own mean and standard deviation, to use as input into the LSTM, according to the activation required, which is a significant part of the working architecture. All parameters were selected after running cross-validation. Additionally, we used a shuffle buffer size of 50, randomly sampling elements from this buffer, replacing the selected elements with new elements but maintaining order. The LSTM outputs are the next Wednesday future prices for the four stocks of our portfolio. We used stock prices to determine each weekly stock's return because the efficient frontier works with returns. We must clear the return using the stock's price forecasting and the stock's price previously. We worked separately with each stock's LSTM neural network and then assembled them, generating a value for the correlation that it observed, this fact being the most remarkable in the methodology.

We used the results of the neural network to return and build the efficient frontier. The efficient frontier is the set of portfolios that satisfy the condition that no other portfolio exists with a higher expected return but with the same risk. To build the efficient frontier, we needed both the return and the risk; we used the LSTM return but to calculate risk, we use the standard deviation return in each case. Additionally, we wanted to select one portfolio in this frontier and compare it with a Navy model.

The navy model selected was the traditional mean-variance efficient frontier, it will work as well as our benchmark. In our case, we wanted to compare the two different models at the same level of risk; we chose the first portfolio at the defined level of risk and invested in it; every week, we carried out the same procedure, and the portfolio was rebalanced to reinvest. After two months, we compared with two metrics, the returns and the Sharpe

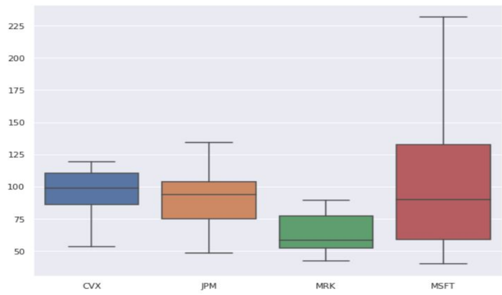
Ratios for both portfolios. Sharpe Ratio is used to help investors understand the return of an investment compared to its risk, and is is the average return earned in excess of the risk-free rate per unit of volatility or total risk.

In this paper, the most remarkable effort was using machine learning algorithms to predict stock prices. Instead of standard statistics, LSTM like all ML algorithms is not limited by assumptions, which allows the data to speak for itself. The non-linear mapping capabilities and the non-assisted estimation of the structural model's parameters of the neural networks are advantageous for its application in the prediction of stock's future returns. This is different a standard statistical time series algorithms, where a separate model is developed for each series. However, the strength of this architecture is produced by individual stock prices with their own LSTM neural network and also use the output of the last layer to train in a single model.

5 Results

5.1 Data analysis

To better understand data and decide what we wanted to forecast, we created some descriptive statistics for all series. First, we plotted a Box-plot (Figure 4) where we could observe the prices that each stock had taken and generated a table with this information. We can see that the volatility of MSFT is the largest, while MRK shows more stability during this period.



	CVX	JPM	MRK	MSFT
count	261.000000	261.000000	261.000000	261.000000
mean	97.462471	88.633682	63.436968	99.028798
std	14.290232	22.094417	13.315481	46.933302
min	53.482479	48.270348	42.465824	40.223278
25%	86.429588	75.346542	52.521008	58.920860
50%	99.100410	93.937920	58.256351	90.286751
75%	110.460342	103.899887	77.357368	132.471863
max	119.106483	134.515091	89.423592	231.649994

Figure 4: Return Boxplot

Nevertheless, the volatility information is not enough; we want to understand all stocks' correlations in order to know whether if one variable changes in value, then the other variable also tends to change in a specific direction. To invest, we are looking for a diversified portfolio where the lower the correlation between the two stocks, the greater the benefit of diversification. The correlation is stronger between MSFT and MRK and weaker between JPM and CVX, as shown in Figure 12 in appendix I. Figure 13 in appendix I shows the distribution of all stocks, with mean zero, and some significant correlations between them.

In neural network applications, the input data is usually normalized or standardized into a range according to the neurons' activation function. Thus, the stock price was standardized to the range [-3, 3] using data training, and we used the same mean and variance to standardize the test set. We plotted the four standardized stocks with their own mean and variance (Figure 5).

Figure 5: Standardize price

5.2 Results Validation Data

This section shows the obtained results. We selected the most relevant parameters in the neural network through cross-validation. After selecting the neural network, we ran the stocks' returns for the months of August and September 2020. With this information, we built an efficient frontier. We selected a portfolio at a risk level and compared the neural network model's results with the benchmark model which will be compared in the next section. There is a better Sharpe ratio and profitability in the neural network model than in the benchmark model.

5.2.1 Selecting the LSTM neural network's parameter

This section applies the cross-validation technique to select the learning rate, an important variable in the LSTM model. The metric used to select the best model was mean absolute error. We used stochastic gradient descent in our method to optimize the lost function in the neural network, with a momentum of 0.9 and an epoch of 200. We used 40 to the LSTM layer, it was backtesting in the M4 forecasting competition, and show us a consistency result (Smyl, 2020). But we used cross-validation to select the learning rate because it is the most critical hyper-parameter control, since it controls how much the model change has in response to the estimated error each time the model weights are updated. Choosing the learning rate is a challenging task since a value too small may result in a lengthy training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process. For this reason, we perform a tuning process with different learning rates (Figure 15 in Appendix II) and selected the optimal.

The learning rate will interact with many other aspects of the optimization process, and the interactions may be non-linear. Nevertheless, in general, lower learning rates will require more training epochs, and conversely, larger learning rates will require fewer training epochs. We created a line plot (Figure 14 in Appendix II) to investigates how the learning rate impacts the model's rate of learning and learning dynamics. For this, we plot the loss in the training and a mean absolute error (MAE). We wanted to minimize overtraining epochs during training where the learning rate might be too large via oscillations in the loss.

5.2.2 Model vs Benchmark performance

We understand how Stocks prices from August 1 to September 30, 2020 are compared with the forecast obtained in the LSTM (Figure 16 in Appendix II). After we got stock prices for each company one from the LSTM, we calculated simple net stock returns. We saw acceptable behavior in the results of the LSTM when stock prices were forecasted. They were softer than real stock prices when compared in the same period. However, as they follow the increasing or decreasing trend, where the trend is remarkable, forecasting presents better behavior.

The benchmark is a pre-determined standard or point of reference against which other things, people, costs, time, or activities can be measured. It is an achievable standard, but when it is not achieved, it could deem the work in question unsatisfactory. Thus, we used a comparative between mean returns for the evaluation of methodology. It is calculated by adding the product of all possible return probabilities and returns and comparing them with the LSTM neural network results: in other word placing them against the weighted average of the sum with the returns is calculated by the LSTM neural network. In the next plots (Figure 6 and Figure 7), the difference between the forecasting means and LSTM with the real return can be observed. The return forecasting through means presents very static behavior. In contrast, the LSTM return is still static but tries to follow the real behavior, and although different in magnitude, it is very similar in trend.

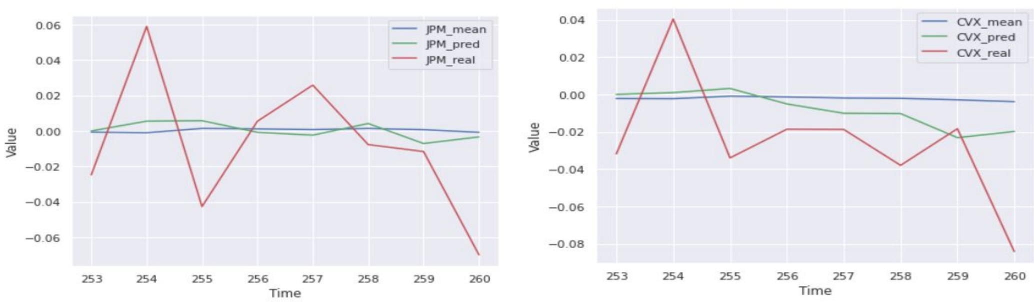


Figure 6: Stock's Return

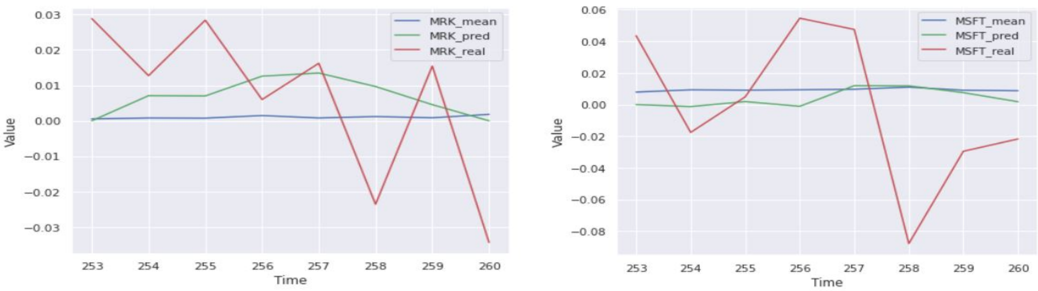


Figure 7: Stock's Return

Between August 1 and September 30, 2020 (Figure 8), it is evident how the portfolio that got through the LSTM neural network with a level risk of 40 % has a better performance than the benchmark, generating in the two months an alpha of 11.8 %,whereas the return of the LSTM was -3.18%, while the benchmark was -14.73 3 %. To clarify, we annualize all returns and volatility to be comparable to each other. In the axis y, we could see the portfolio value if we had invested 1 usd in the first week, and in the axis x, the two months represented in 8 weeks.

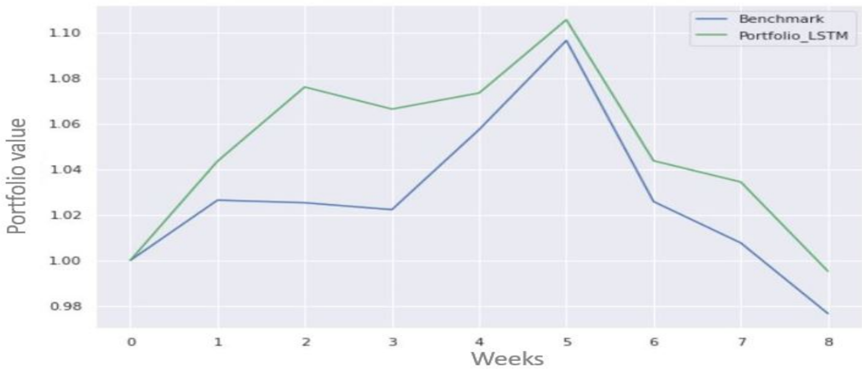


Figure 8: Portfolio Return: Ago 20 - Sep 20

Figure 9 shows the results obtained in terms of percentage for each stock for the last portfolio invested in each one. The LSTM gives a higher percentage to MRK over MSTF, and in both cases, neither allocates any portion to CVX. The results shown in the next table on different dates are the returns obtained for each portfolio obtained and the same level of risk and Sharpe ratio. We assembled the four stocks' neural networks to know that each one was learned from the other stocks' neural networks and that is one of the reasons the LSTM neural network learning performed well and thus helped them forecast better stock prices.

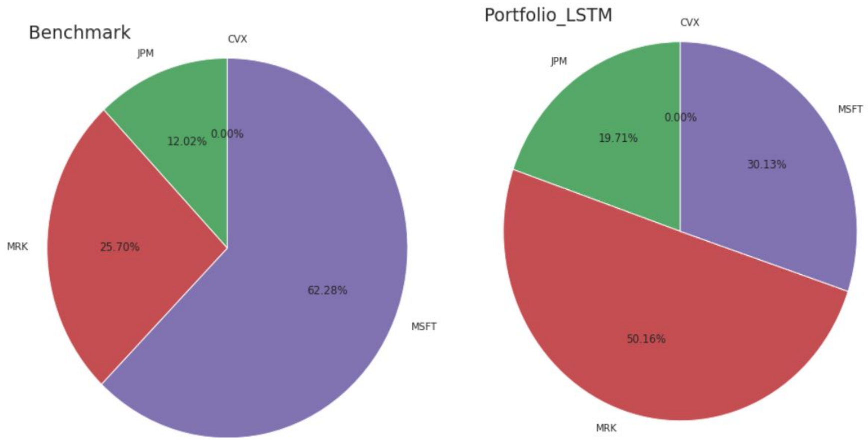


Figure 9: Allocation Portfolio

5.3 Evaluation Model: Cross Validation

To evaluate the neural network's learning behavior, we used cross-validation, which was carried out by evaluating the results in different periods. Using machine-learning models requires a training set and a test set in the training evaluation procedure. We used time-based splitting in a temporal series to provide a statistically robust model evaluation and to best simulate real-life scenarios. In this case, we used time-based cross-validation, a method taken from the time-series field which forms a type of "sliding window" training approach (Figure 10). We have 253 days to train and 8 days to test in each cross fold.

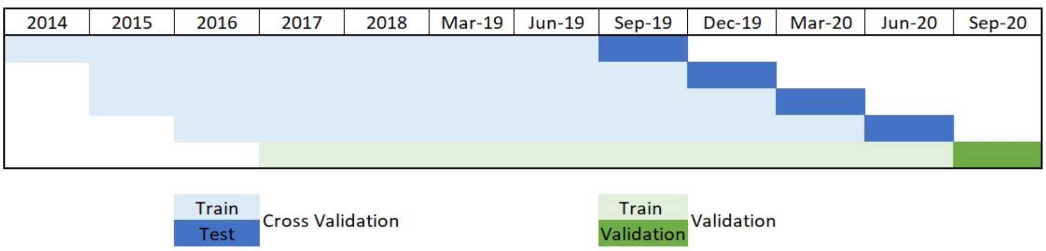


Figure 10: Train Test Set

The Table 1 shows the different periods in which the LSTM neural network was tested from August to September 2019, always training the network for 5 years and testing it after two months and the portfolio was rebalanced every week on Wednesday. We invested according to the portfolio allocation. The results show how all the periods generate higher profitability at the same level of risk and, therefore, a better Sharpe ratio than the benchmark.

Table 1: Results best model against benchmark portfolio.

Metrics	Return		Volatility	Sharpe Ratio	
Date	**Benchmark**	**LSTM**		**Benchmark**	**LSTM**
Ago 19 - Sep 19 fig:17	11.13%	20.2%	40%	0.28	0.51
Nov 19 - Dic fig:18 19	66.35%	82.3%	40%	1.66	2.06
Feb 20 - Mar 20 fig:19	-81.31%	-75.98%	40%	-2.03	-1.90
Mar 20 - Jun 20 fig:19	24.52%	34.03%	40%	0.61	0.85
Ago 20 - Sep 20 fig:8	-14.7%	-3.18%	40%	-0.37	-0.08

The figures in appendix II show the LSTM and benchmark portfolios' behavior between the two months used to test for each run on the different dates and their behavior. In general, this shows good performance, but the LSTM portfolio always has better performances. Inside the LSTM neural network architecture, we have a layer where the 4 stocks converge, and the prices are delivered to consider the relationships among the four stocks. After this, the same risk level is run and calculates the portfolios' return and where the results are straightforward.

The best result was between November and December 2019, while the worst was between May and June 2020, although it still shows a better return than the benchmark. A trader who took these positions vs. taking them with the traditional mean-variance frontier would have obtained a better return and a better Sharpe ratio. The findings of this study clearly show that, in general, the efficient frontier in all the runs had better performance. Since the LSTM does not retrain in the two months in which it was testing, it does not always win, but it shows how the portfolio has a better Sharpe ratio.

6 Conclusions and future work

- Selecting the optimal portfolio continues to be a challenge for traders. They are always looking for a portfolio that optimizes the relationship between return and risk. Since there is an uncertain future and all estimations are with samples but not including the entire population, alternatives such as deep learning, as shown in this work, help reinforce improvements in forecasting the return and select a better portfolio to invest.
- The LSTM networks work for the projection of time series, and as in terms of profitability, it shows that it tries to follow the real stock price closer at least in its behavior than what it does to calculate the average.
- The LSTM neural network assembly for four stocks allowed them to generate better stock price forecasting so that the key to their success was that they were given an extensive representative data set, in this case the information for all four stocks.
- In each evaluation window, the LSTM neural network beat the benchmark. Never-

theless, the model has retraining every two months; we must check whether a higher training frequency can improve the LSTM performance.

- In this work, we demonstrated that we get more efficient portfolios at the same risk but modify the expected return. However, in the future, it will be possible to evaluate not only changing the return but also the risks, taking them as the error generated by the stock prices' neural network forecasting. Additionally, we will evaluate not only with four stock, if not incorporating more stock prices with provide more information.
- we could not only used neural network (LSTM) to predict stocks' returns, if not we can change the risk measure, based on the prediction errors, that have the same statistical foundation of the mean-variance model for future work. In this case, we will change covariance matrix as well.

References

CVX data. (n.d.). https://finance.yahoo.com/quote/CVX?p=CVX&.tsrc=fin-srch/. (Accessed: 2010-09-30)

Fisher, T., & Krauss, C. (2017). Deep learning with long short-term memory networks for financial market predictions. *FAU Discussion Papers in Economics*.

Freitas, F. D., De Souza, A. F., & De Almeida, A. R. (2009). Prediction-based portfolio optimization model using neural networks. *El Sevier*, *72*, 2155-2170.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm. *Neural Computation*, *12*, 2451–2471.

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *In Acoustics, speech and signal processing*, 6645-6649.

Hansen, R., & Nelson, J. (1997). Neural networks and traditional time series methods:a synergistic combination in state economic forecasts. *IEEE Transactions on Neural Networks*, *8*, 863-873.

Heaton, J., Polson, N., & White, J. (2016). Deep learning in finance.

Jeon, S., Hong, B., & Chang, V. (2018). Pattern graph tracking-based stock price prediction using big data. *Future Generation Computer Systems*, *80*, 171-187.

JPM data. (n.d.). http://finance.yahoo.com/quote/JPM?p=JPM&.tsrc=fin-srch/. (Accessed: 2010-09-30)

Kohara, K., Ishikawa, T., Fakuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. *Intelligent systems in accounting, finance and management*, *6*, 11-22.

Lee, M. S., Ahn, C. H., Kwahk, K. Y., & Ahn, H. (2018). Stock market prediction using convolutional neural network that learns from a graph. *International Journal of Business and Economics Engineering*, *5*.

Li, X., Xie, H., & Wang, R. (2016). Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications*, *27*, 67-78.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *The Journal of Economic Perspectives*, *17*, 59-82.

Markowitz, H. M. (1952). Portfolio selection. *Journal of Finance VII*, *1*, 77-91.

MRK data. (n.d.-a). https://finance.yahoo.com/quote/MRK?p=MRK&.tsrc=fin-srch/. (Accessed: 2010-09-30)

MRK data. (n.d.-b). https://finance.yahoo.com/quote/MSFT?p=MSFT&.tsrc=fin-srch. (Accessed: 2010-09-30)

Mullainathan, J., S.and Spiess. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, *31*, 87-106.

Phi, M. (2018). *Illustrated guide to lstm's and gru's: A step by step explanation.* https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Po$_C hang, K., \&Pin_C heng, L.$ (2008). Resource allocation neural network in portfolio selection. *Expert Systems with Applications*, *35*, 330-337.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *30*, 75-85.

White, H. (1988). , economic prediction using neural networks: the case of ibm daily stock returns. *Proceedings of the IEEE International Conference on Neural Networks*, 451-458.

Appendix I: Statistics Data

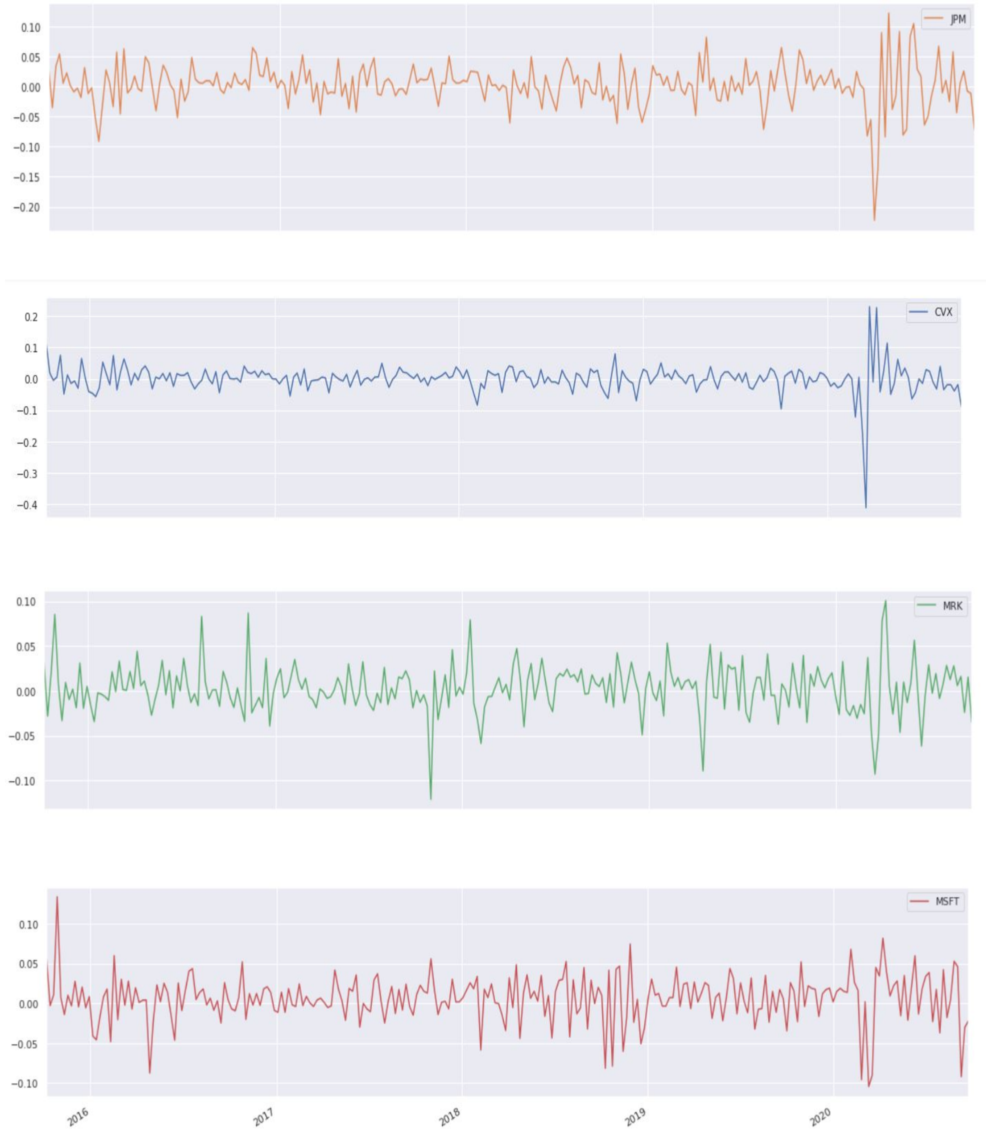


Figure 11: Stock's Return

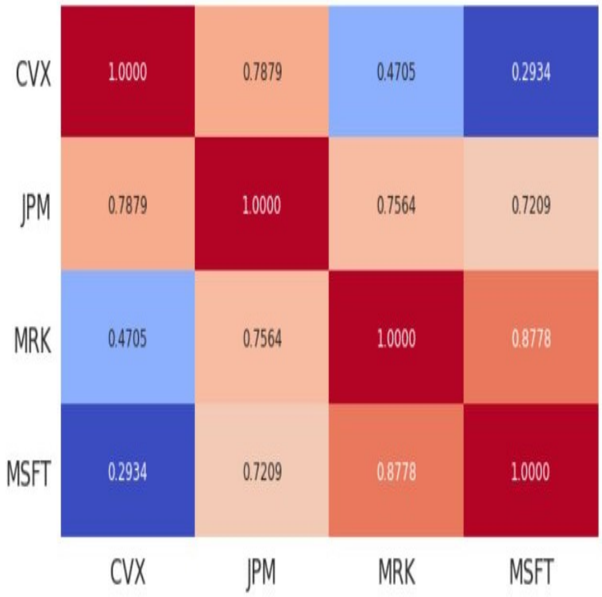


Figure 12: Correlation

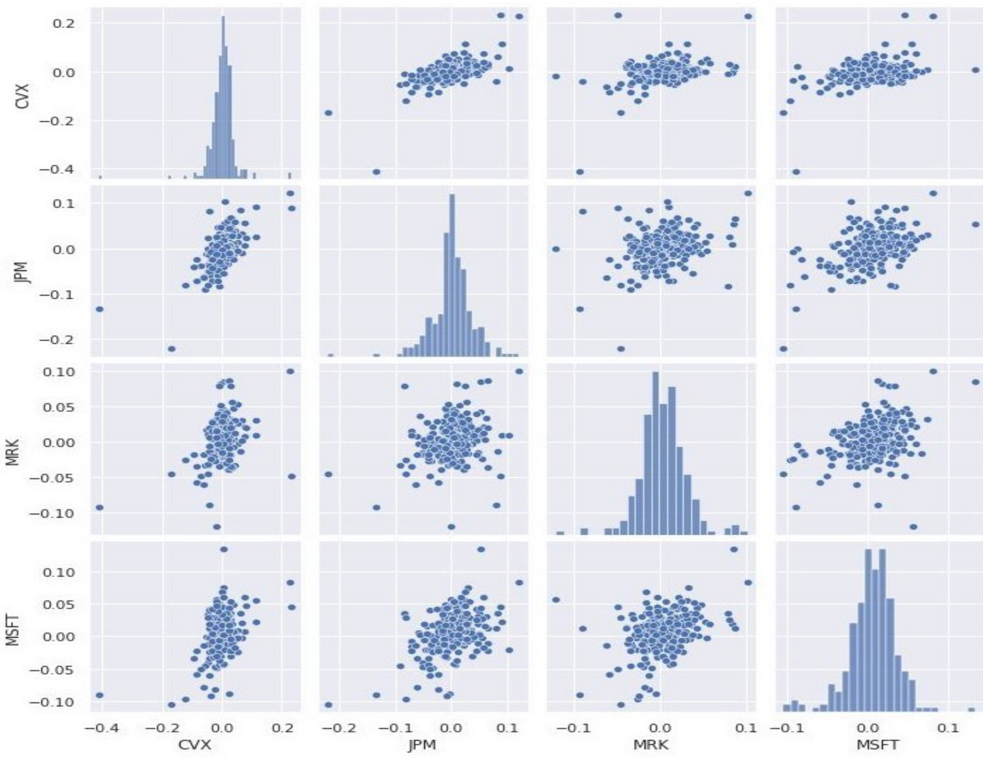


Figure 13: Distribution

Appendix II: Evaluation Model

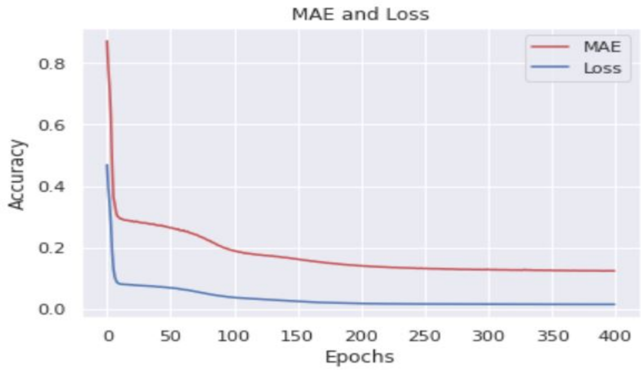


Figure 14: Loss/MAE vs Epoch

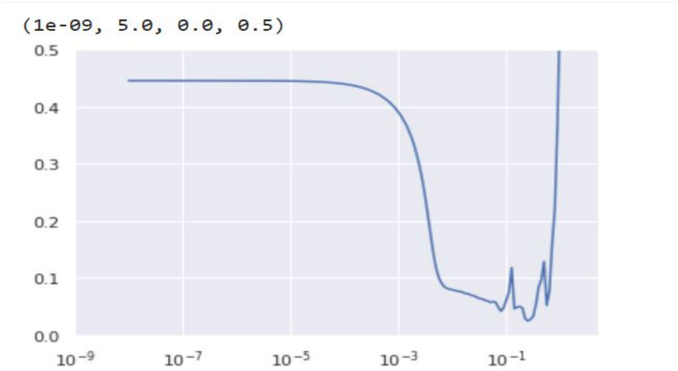


Figure 15: Tuning Learning Rate

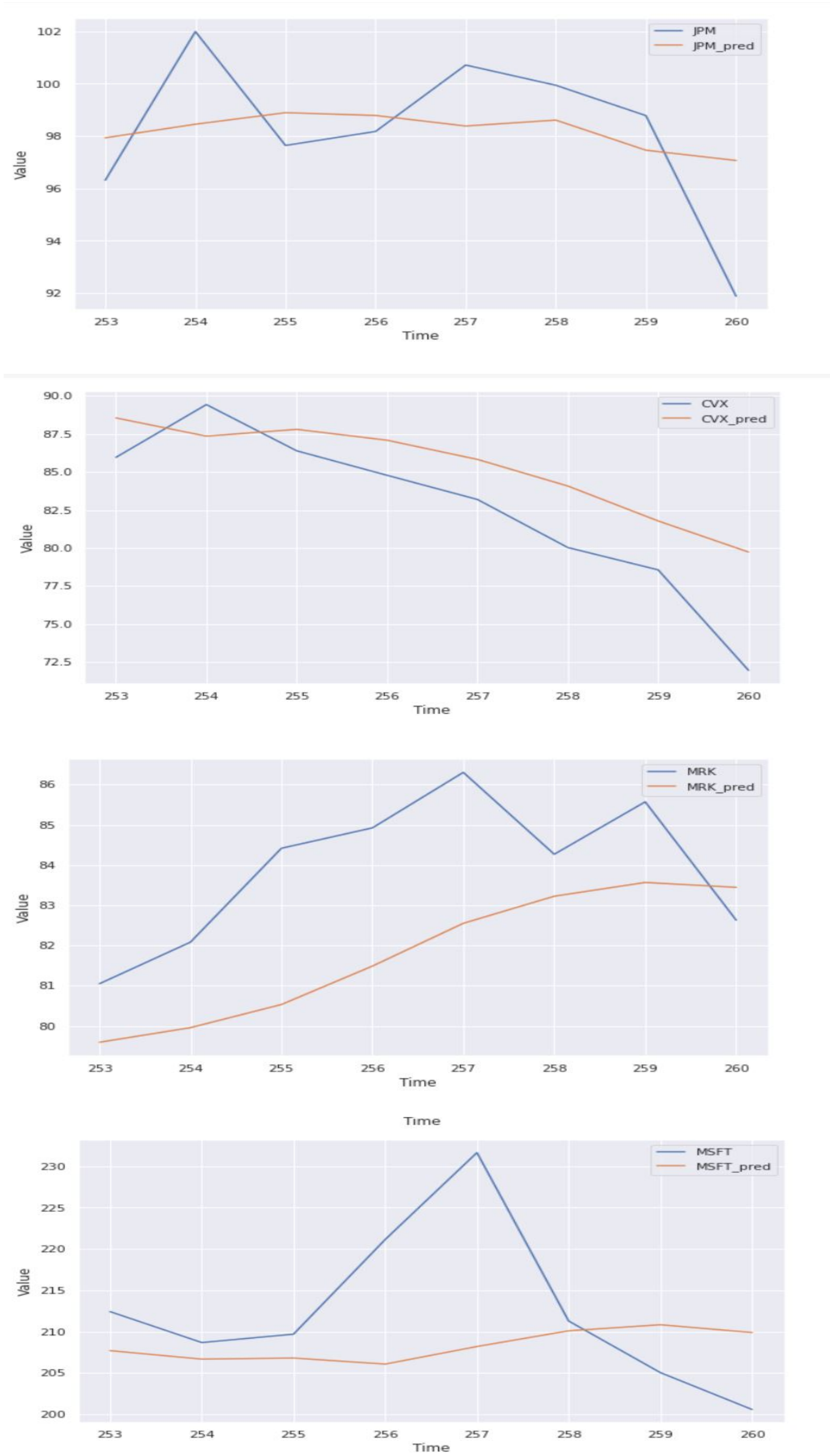


Figure 16: Stock's Prices

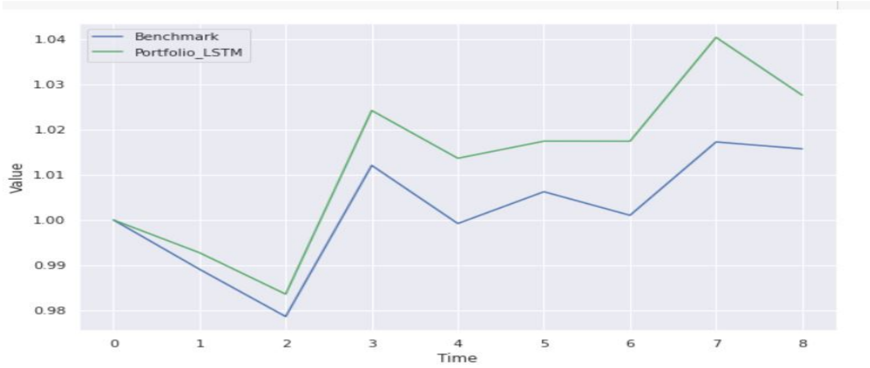


Figure 17: Ago 19 - Sep 19

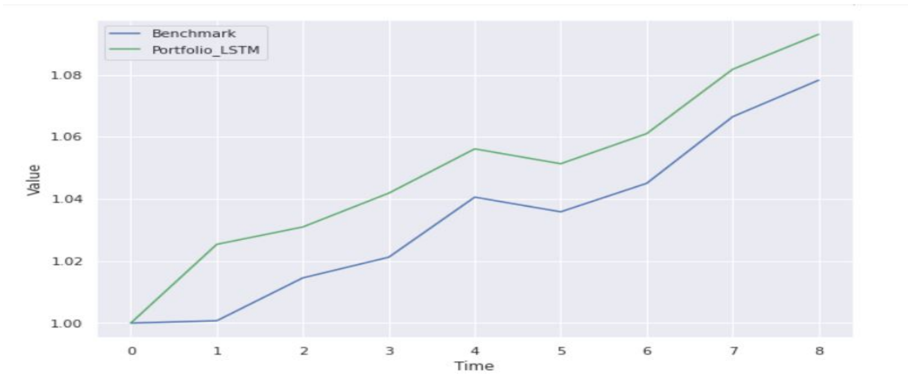


Figure 18: Nov 19 - Dic 19

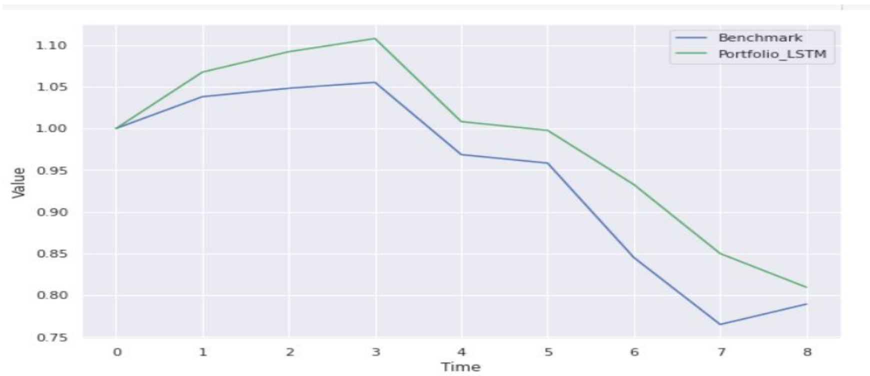


Figure 19: Feb 20 - Mar 20

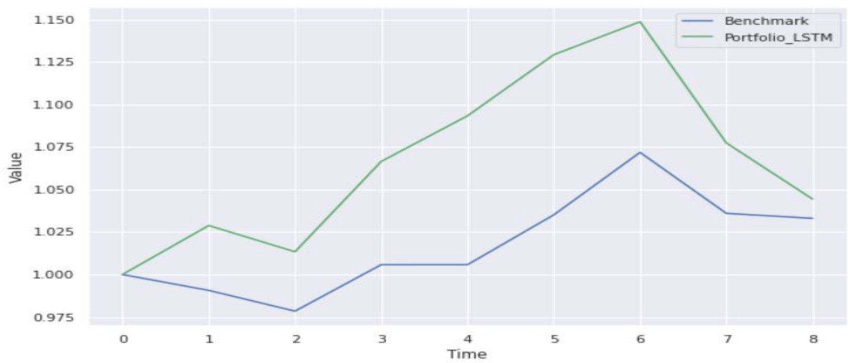


Figure 20: May 20 - Jun 20